Overview: 1. Relationships between classes  
   (e.g. P, NP, PSPACE...)  
2. TQBF ∈ PSPACE  
3. LADDER, DFA ∈ NPSPACE  
4. LADDER, DFA ∈ PSPACE  

> Savitch's Theorem  
PSPACE = NSPACE
Relationships between $P$, $NP$, $coNP$, $PSPACE$, $coPSPACE$

$NPSPACE$

$EXPSPACE \neq P$

$EXP \neq \text{TIME}$

$NPSPACE = PSPACE = coPSPACE$

$TQBF = \text{LADDER DFA}$

$coNP$

$SAT$

$\text{TAUTOLOGY}$

$NP$

$\text{SAT}$, $3$-$\text{SAT}$

$\text{HAMPATH}$

$P$

$\text{PATH}$

$\text{MOD-EXP}$

$P = NP$?  
$P = PSPACE$?  
If $P \neq NP$, $NP = PSPACE$? and so on

- $PSPACE = coPSPACE$
  - $PSPACE$ is deterministic.
  - So everything in $coPSPACE$ is decidable in polynomial space deterministically.

- $NPSPACE = PSPACE$
  - Not obvious, Switch's Theorem.
TQBF

Def: quantified Boolean formula (QBF) is a Boolean formula w/ quantifiers (\( \exists, \forall \)). All variables in the formula must be quantified.

QBF is true or false.

Def: TQBF = \( \exists \phi \) / \( \phi \) is QBF that is True.

Ex: \( \exists x [x] \) is this true/false?
True. Set \( x = \text{true} \).

\( \forall x [x] \) is this true/false?
False. Set \( x = \text{false} \) \( \rightarrow \) false result.

\( \exists x_1 \forall x_2 \exists x_3 \forall x_4 \ldots \)
\( \phi = \forall x \exists y [ (x \lor y) \land (\neg x \land y) ] \)
\( \phi \in \text{TQBF} \)
\( x = \text{true}, \) pick \( y = \text{false} \) \( \rightarrow \) T
\( x = \text{false}, \) pick \( y = \text{true} \) \( \rightarrow \) T
**Theorem:** TQBF $\in$ PSPACE.

**Idea:** Use recursion

- Suppose $\phi = \exists x \cdot \psi$
  - Just try setting $x = \text{True}$ in $\psi$ and $x = \text{False}$ in $\psi$. If either evaluates to $\text{true}$, then $\phi$ is $\text{true}$, so accept.
  - Otherwise reject.

- Suppose $\phi = \forall x \cdot \psi$
  - Try setting $x = \text{True}$, $x = \text{False}$.
    - If both evaluate to $\text{true}$, accept.
    - Otherwise reject.

Recurse on $\psi$.

- We set $x$ to a boolean value.
- $\psi$ is just another QBF, but now it has $x$ set.
- So we can pass in $\psi$ (with $x$ set) back into our formula.
If $\phi$ has no quantifiers (so it has no variables), then it's just a boolean value, so output that value.

Space complexity analysis:

When we do the recursion, we want to use the same space (so don't copy the formula to new space), $\text{input } | = n$.

$\exists x_1, \exists x_2, \forall x_3 \ldots x_1 \lor x_2 \lor \overline{x_3} \ldots$

Assignments:

<table>
<thead>
<tr>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T v F</td>
</tr>
</tbody>
</table>

Formula
LADDER$_{\text{DFA}} \in \text{NPSPACE}$

**Def:** \[ \text{LADDER}_{\text{DFA}} = \{ \langle B, u, v \rangle \mid B \text{ is DFA, } L(B) \text{ has the ladder } y_1, \ldots, y_k, \text{ where } y_1 = u, \ y_k = v \} \]

**Proof:** Here's an algorithm:

1. Start by string $y = u$. Let $n = |u|$. 
2. Repeat at most $\lfloor \frac{1}{\varepsilon} \rfloor$ times.
   2a. Nondeterministically change $y$ at 1 character.
   2b. Check if $y = v$. If so, accept.
   2c. Check if $y \in L(B)$. Reject if $y \notin L(B)$.
3. Reject.

This uses linear space. \[ \text{LADDER}_{\text{DFA}} \in \text{NPSPACE} \]
LADDER

Proof idea: Do recursion, but less recursion than a naive way by using binary search.

Subproblem: BOUNDED-L LADDER DFA

Claim: B-L can be solved w/ binary search.

Recursion depth \( \Theta(\log b) \).

At every level of recursion, we use \( \Theta(n) \).

Decide LADDER DFA, for a given input \( \langle B, u, v \rangle \), by passing \( \langle B, u, v, 1 \leq l^m \rangle \) into B-L. This uses \( \Theta(n) \cdot O(\log b) \) space
\[ = \Theta(n) \cdot O(\log 13^m) \]
\[ = O(n) \cdot O(n) = O(n^2) \]

So, LADDER$_{DFA} \in \text{PSPACE}$. 

\[ \text{NPSPACE} = \bigcup_{k} \text{languages that can be decided in } O(2^kn^k) \text{ space, nondeterministically} \]